Predicting Bank Defaulters using Machine Learning

By Jack Skelton

# Tables of contents

[Tables of contents 1](#_Toc195443066)

[Table of figures 2](#_Toc195443067)

[Project aim 3](#_Toc195443068)

[Project expectations 3](#_Toc195443069)

[Literature Review 6](#_Toc195443070)

[Methodologies researched 12](#_Toc195443071)

[Machine Learning Approach 12](#_Toc195443072)

[Heuristic Approach 13](#_Toc195443073)

[Conceptual System 14](#_Toc195443074)

[Methodologies comparison 15](#_Toc195443075)

[Review of technologies 15](#_Toc195443076)

[Machine Learning models chosen 16](#_Toc195443077)

[Methodology chosen/Framework 16](#_Toc195443078)

[Data 17](#_Toc195443079)

[Missing values 17](#_Toc195443080)

[Finding outliers 21](#_Toc195443081)

[Results 23](#_Toc195443082)

[Conclusion 33](#_Toc195443083)

[Appendix 33](#_Toc195443084)

[Risk register 33](#_Toc195443085)

[Bibliography 34](#_Toc195443086)

[Use of Generative Artificial Intelligence (GAI) Statement 35](#_Toc195443087)

[Monthly reports 36](#_Toc195443088)

[Github 46](#_Toc195443089)

[Video 47](#_Toc195443090)

# Table of figures

[Figure 1 photo shows the results for the Disease prediction using machine learning 9](#_Toc195464640)

[Figure 2 chart shows the models used in discussed papers 10](#_Toc195464641)

[Figure 3 shows the performance of the algorithms using the machine learning approach Methodology 12](#_Toc195464642)

[Figure 4 machine learning steps 13](#_Toc195464643)

[Figure 5 Heuristic method approach steps 14](#_Toc195464644)

[Figure 6 diagram of conceptual system used 15](#_Toc195464645)

[Figure 7 table of software comparisons 16](#_Toc195464646)

[Figure 8 code for missing values 17](#_Toc195464647)

[Figure 9 picture of dataset showing missing data 18](#_Toc195464648)

[Figure 10 code for making null values 18](#_Toc195464649)

[Figure 11 dataset showing null values 18](#_Toc195464650)

[Figure 12 code to find missing values 19](#_Toc195464651)

[Figure 13 output of md.pattern code 19](#_Toc195464652)

[Figure 14 KNN method code 19](#_Toc195464653)

[Figure 15 photo of dataset showing null values removed and filled 20](#_Toc195464654)

[Figure 16 md.pattern output after data imputation 21](#_Toc195464655)

[Figure 17 code used to save changed dataset to an excel file 21](#_Toc195464656)

[Figure 18 code for finding outliers 21](#_Toc195464657)

[Figure 19 output showing outliers 22](#_Toc195464658)

[Figure 20 implementation of logistic regression 23](#_Toc195464659)

[Figure 21 logistic regression accuracy 24](#_Toc195464660)

[Figure 22 implementation of KNN 25](#_Toc195464661)

[Figure 23 KNN accuracy 26](#_Toc195464662)

[Figure 24 implementation of naive Bayes 27](#_Toc195464663)

[Figure 25 Naive Bayes accuracy 28](#_Toc195464664)

[Figure 26 implementation of random forest 29](#_Toc195464665)

[Figure 27 random forest accuracy 30](#_Toc195464666)

[Figure 28 implementation of decision tree 31](#_Toc195464667)

[Figure 29 decision tree accuracy 32](#_Toc195464668)

[Figure 30 table of all accuracy scores 32](#_Toc195464669)

[Figure 31November monthly report 38](#_Toc195464670)

[Figure 32 December monthly report 40](#_Toc195464671)

[Figure 33 January monthly report 42](#_Toc195464672)

[Figure 34 February monthly report 44](#_Toc195464673)

[Figure 35 March monthly report 46](#_Toc195464674)

[Figure 36 photo of github page 46](#_Toc195464675)

[Figure 37 Photo of dashboard 46](#_Toc195464676)

# Project aim

My project aim to “develop an AI using the coding software to make predictions on a customer account to see if it is default.” This project would allow banking organizations to predict which accounts are at risk of becoming default from missing payments, this will be done to allow the banks to minimize the losses they would have received without the predictions. I will be developing my coding skills throughout this project learning on how to implement different machine learning modules into python scripts by creating the code for them. This would be an appropriate challenge for level 6 since I will be using machine learning which may not have complex code but will have great understanding on what happens in the process which will make my understanding better for my studies since I will be studying machine learning in semester 2.

# Project expectations

Investigation

In this investigation I will be creating 5 different scripts which will include 5 different machine learning modules. These will include these individual modules Naive Bayes, Decision Tree, Random Forest, Regression, KNN. I will then take a data set from archive.ics.uci.edu to get some banking details. I will then use this data set and put it into scripts to teach the AI to predict if an banking account will be at the risk of becoming default. I will do this by looking at the dataset for banking details to look at the past payment and other data to make predictions to see if a user is going to miss a payment to allows the banks to find out what accounts are at risk. I will then compare the different machine learning modules predictions to identify what machine learning modules makes the best predictions and this python script will be my final product. I will need to do a video at the end of the project to display how the code on the different scripts works and how the predictions are made, I will also have to show how I identified which machine learning module makes the best predictions. The data set I will be using will be Default of Credit Card Clients (see appendix B for link or click data set name).

Evidence good practice

I will evidence my findings by using a report documenting all my findings on the machine learning predictions and I will also be making a video to record my findings on what the predictions where and what the best machine learning module was to predict the risk of an account becoming default. I will also produce a monthly project report to document what has happened every month during the production module, I will also produce a use of generative AI which will document the use of AI like ChatGPT throughout the production module which will show what queries I have asked an AI to get some answers. All this would show good practice and allow the project to be successful. I will also have to stay within the laws like the computer misuses act since I will be using a dataset from someone else so the data is not mine, thus I will need to be careful about what I use the data for to not breach any laws or morals.

Appropriate evaluation

An appropriate evaluation of this project can be broken down into several aspects: relevance, importance, cost and quality. My project is relevant to modern times since I will be using and creating some machine learning models that is a part of an industry that is growing and most things now have some connections to AI, so this would be good to do since I will help me find a career after university with important skills. My project will only cost me time, since no money is needed to complete my project since the only bit of software I am using is free, though this project will cost me time since I will need to spend a lot of time researching the different machine learning models and creating and testing the scripts. The quality of the project is up to me and is impacted by how much time is spend on the project, having a good quality product would mean that the project was a success and that I could then use the skills developed to help me in the future and would be a nice talking point in an interview. The importance of this project is quite high since if this is a successful project then the project produced out of the project would be helpful to the banks and would allow them to reduce their losses and make sure they could update any users for the risk that they are approaching with their account so they done become bankrupt and have no outstanding debts.

A MoSCoW type requirements analysis or equivalent

|  |  |  |  |
| --- | --- | --- | --- |
| **MUST**  (40-50%) | **SHOULD**  (50-70%) | **COULD**  (70%+) | **WON’T**  (To provide clarity on what won’t be included) |
| * Create 5 different machine learning models * Test ai for predictions * Evaluate which ai is best for predction * Document process | * Create ai models in R script * Test ai for default risk * Only submit one machine learning model for my product | * Have hybrid machine learning models to predict crime rate * Use data other than classification | * More than 5 machine learning models * Using different programming languages |

# Literature Review

|  |  |  |  |
| --- | --- | --- | --- |
| **Research Objective** | **Methodology/Tools Used** | **Results** | **Paper Reference Number** |
| Predict TBM operation parameters using machine learning | Decision tree | Proposed a BAS algorithm  Improved rationality  Reduced uncertainty  Better predictions | Wang, Y. et al. |
| Machine Learning Models for Predicting the Ammonium  Concentration in Alluvial Groundwaters | Classification  Regression  Baseline models | Deep Neural Network outperformed other model in predictions, this was shown with the plots comparison | Perović Marija et al. |
| Prediction of metastatic pheochromocytoma and paraganglioma | Decision tree | Highest performance is achieved by SVM and LogReg, SVR and RR models | López Steinmetz, L.C. et al. |
| Machine learning models predict the emergence of depression in Argentinean college students during periods of COVID-19 quarantine | Dual methodology | Shows pre existing depression and anxiety  Machine learning models show potential at risk students | Pamporaki, C. et al. |
| Open AccessEditor’s ChoiceReview  Flood Prediction Using Machine Learning Models: Literature Review | Flowcharts | Loads of articles reviews and studies were analyses  There was an effective use of machine learning models | Mosavi, A., Ozturk, P. and Chau, K.-w. ( |
| Designing Disease Prediction Model Using Machine Learning Approach | K-Nearest neighbor, Convolutional neural network | The system shown in this method worked on showing the disease risk prediction  Machine learning models have a good comparison showing accuracy and processing time of both models  the system shown has the potential to show large scale medical data effectively | Dahiwade, D., Patle, G. and Meshram, E. |
| Disease Prediction using Machine Learning Algorithms | Disease Tree Classifier  Random forest Classifier  Naïve Bayes Classifier | This article shows how accurate the algorithms is at 95% using medical classification  This model has some future proof since the more medical data releases the more accurate the algorithm will be | Grampurohit, S. and Sagarnal, C. |

|  |  |  |
| --- | --- | --- |
| **Previous work** | **Limitations** | **Proposed System**  **(How it will cover those limitations** |
| Machine learning models predict the emergence of depression in Argentinean college students during periods of COVID-19 quarantine | Further studies are needed before clinical implementation | Make sure that the data set is proper and make sure there are no empty data fields |
| Open AccessEditor’s ChoiceReview  Flood Prediction Using Machine Learning Models: Literature Review | Machine learning was in early development  There are some data limitations in this method  Some machine learning models require good optimization for it to work | Machine learning is still growing but in a better place now  Check dataset for any data limitations  Make sure I am using a suitable software for the coding |
| Designing Disease Prediction Model Using Machine Learning Approach | They only used two different machine learning models  The system proposed depends on the data sets quality and size | I will be using multiple different machine learning models  Make the product able to handle different data sets |
| Disease Prediction using Machine Learning Algorithms | This study only uses 3 models limiting the potential performance  The model relies on quality and quantity of the data  Matrix are useful but do not fully capture model effectiveness | Will be using more models to predict  Make the product handle more data sets  Look at other metrics like precision or recall |

A screenshot of a computer

Description automatically generated

Figure photo shows the results for the Disease prediction using machine learning

|  |  |  |
| --- | --- | --- |
| **Paper types** | **Methodology** | **Year** |
| Article | Decision tree | 2023 |
| Article | Classification  Regression  Baseline models | 2021 |
| Article | Decision tree | 2023 |
| Article | Dual methodology | 2024 |
| Article | Flowchart | 2018 |
| Article | K-Nearest neighbor  Convolutional neural network | 2019 |
| Thesis, Dissertation | Disease Tree Classifier  Random forest Classifier  Naïve Bayes Classifier | 2020 |

Figure chart shows the models used in discussed papers

One of the papers I have had a look at was a paper using the same dataset and a similar project idea, so this paper can be a helpful tool to look at how a past project doing this has taken a approach which I can then look at and see if I can make it better, a key thing I will need to avoid is that I need to make sure my project is not just copying this paper. The paper I am talking about is called “Credit Default Mining Using Combined Machine Learning and Heuristic Approach” this paper was published by Sheikh Rabiul Islam and W. Eberle, S. Ghafoor 2018. The aim for this project was to look at a dataset and predict if an account is default. A default account is an account made by a user that has missed some payments and is closed by the bank. This is helpful since it allows the banks to see what accounts could be at risk of becoming a default account and the probability of a payment in advanced. The paper than states that the reason this is done is to “earlier the potential default accounts are detected the lower the losses” this would then show why it is helpful for banks to make these predictions since it will make the losses through these accounts lower so they don’t lose as much money. Another thing this paper states about the reason for this project is so that it “could investigate and help the customer by providing necessary suggestions to avoid bankruptcy and minimize the loss” this clearly shows that the project is used to help the customer and the bank on both ends since as discussed before it helps the bank lower the losses and shows the customer some helpful things that can be done to help with their payments.

When it comes to this papers methodology they implement 2 different ones. This is a machine learning approach and a heuristic approach. These are then discussed on what these different methodologies are and how they could be used. That machine learning approach is said to be used to compare different machine learning models and see which one is most effective at making their predications, the other one is used to make the predictions since it is used throughout the different machine learning models to make the predictions of what accounts are at risk of becoming a default accounts. This is helpful for my project since it allows me to look a some methodologies and see which would be suitable for my project, these would be helpful since I will be using multiple different machine learning models to compare the different effectiveness of the models to create my final product, and the other one could be used to create the predictions that I will be making in the future.

The results of this paper shows that both methods used in this project were successful at making there predictions. The results shown from this project show that the methods used outperform currently used methods. This shows that there project was a great success and that banking organisations should implement their ideas.

A graph with colorful bars

Description automatically generated with medium confidence

Figure shows the performance of the algorithms using the machine learning approach Methodology

## Methodologies researched

### Machine Learning Approach

Looking at one of the previous papers credit default mining using combined machine learning and heuristic approach they have an approach which was used to determine what machine learning model used was the best for the predications they were trying to make but while looking at the data shown in the paper there was no steps shown what was taken to determine this. So I did some extra research for machine learning approaches and had a look at the steps taken in the method. I found a webpage showing an approach for this method as seen in the image below. This method contains 5 major steps:

1. Collection of data
2. Data cleaning & feature engineering
3. Model building
4. Evaluate
5. Model deployment

Breaking these steps down we can see that in the first step that this is where the user is collecting the data which for me was finding the data set. Step 2 is to look at the data set for any errors in the data like null values. Step 3 is the step done to program the models used for predictions. Step 4 is looking at the results of the each model to determine what was best at making predictions and step 5 is the step where the product is deployed into a working environment.

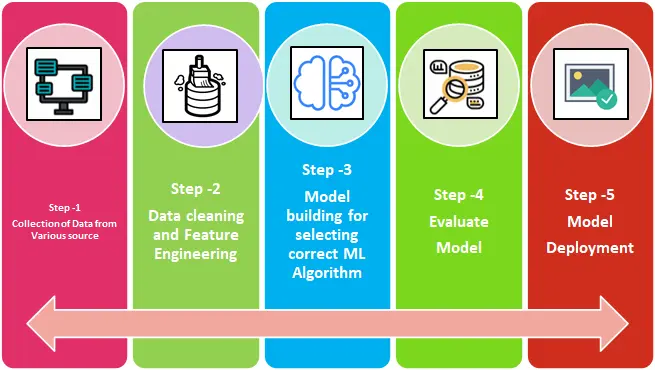


Figure machine learning steps

### Heuristic Approach

The paper looking at prediction of default credit also had a heuristic approach, which is a method that contains two tests a standard test and a customer specific test, these are used to find out which accounts were at the potential risk of becoming a default account. This model was used for two steps one to determine the potential risk of a account becoming a default account by looking at the transitional history. And the other step was to look at the real time transitions of an account to determine a score of the account becoming a default account. Though this paper did not include a diagram of this model when we look at the heuristic approach online we can see what the models looks like in the image below, as we can see that the model has 4 steps to it. The first one is used to look at the problem and understand it by creating a scope, the second one is making a plan to mitigate the problem, step 3 is to carry out the plan created, and the last step is used to evaluate and adjust the method used.

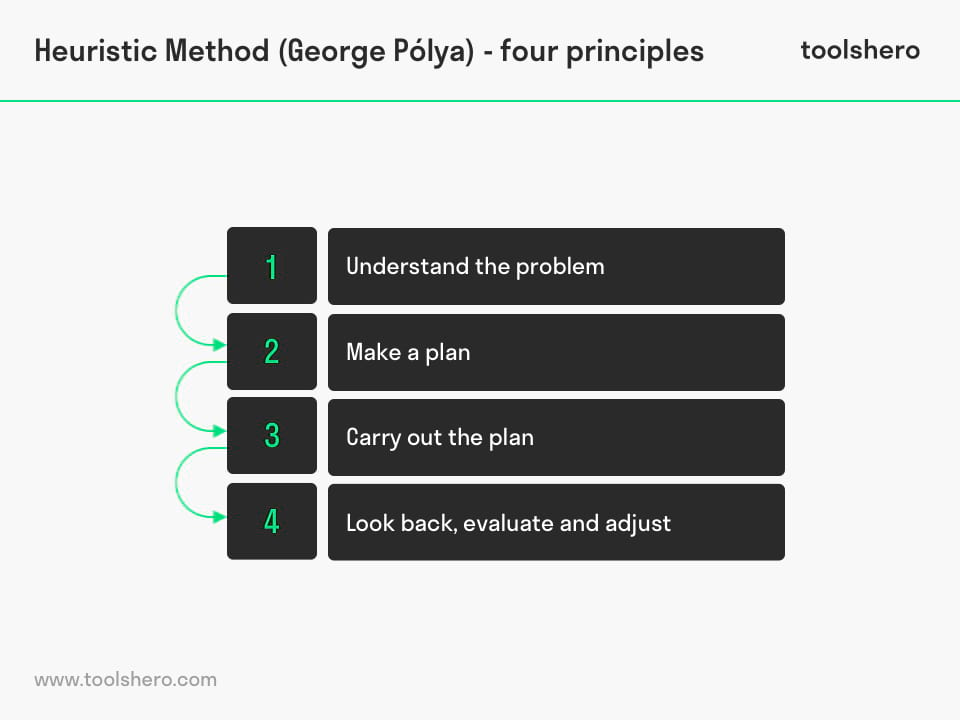


Figure Heuristic method approach steps

### Conceptual System

This method was used in a paper to determine the crop yield based on machine learning models. This model was not discussed in the paper only showing a diagram of how the method was done, this can be seen in the image below. As we can see that the data sourced to the user was then made into a centralized dataset via ETL. Then the models were made and trained using the dataset, the models were also evaluated at this stage of the method to determine what was the best prediction-based model. This was then implemented into a predictions system where it will be feed new data to make predictions based on crop yield.

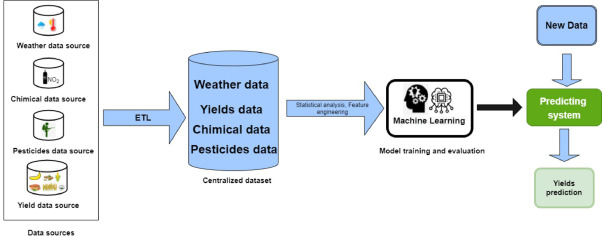


Figure diagram of conceptual system used

## Methodologies comparison

Looking at these methods we can see that they all include machine learning models to make predictions based on a data set and each had an evaluation step to determine what the best model for predictions was for the product. Though there are some differences with the heuristic approach, it is the only one that includes a step which has the user go back and adjust the method used to make the predictions better. The conceptual system was the only method used which had at the end of the model training feeding new data to make predictions and is the only method that has been shown to use ETL in the dataset, though this could be included in the second step of the machine learning approach where it is said that the user must clean the data.

## Review of technologies

These are some of the software that I researched that I could use to complete this project.

|  |  |  |
| --- | --- | --- |
|  | R studio | Python |
| Areas of use | Popular in academic and businesses | Well suited for many types of programs |
| Research, finance and data science | Data science, web development, software development and gaming |
| Language | General-purpose programing language | General-purpose programing language |
| Advantages | * Open source * 19,000 packages * Easier to learn at start * Organized interface * Great for graph making * Many functionalities for data analysis * Great for statistical analysis * Using in machine learning module | * Open source * +300,000 packages * Beginner friendly * English like syntax * High ease of deployment and reproducibility |
| Disadvantages | * More difficult when using advanced functionalities * Slower language * Finding right libraries can be hard | * Poor memory efficiently * Has less libraries for data science * Real time errors * Needs rigorous testing * Visualizations are more convoluted |

Figure table of software comparisons

Comparing the 2 software there are some similarities with them like that there are both general purpose programming languages. But R studio is made for research, finance and data science, which is was I care about in the software for data science since this is was my predictions models are in. python is still a good software to know but for this project the better functionalities for data science like graphs and analysis make this a no brainer in which software to pick for this project.

## Machine Learning models chosen

Chosen models:

1. Naïve Bayes
2. Decision Tree
3. Random Forest
4. Regression
5. K Nearest Neighbors (KNN)

## Methodology chosen/Framework

When it comes to my methodology that I am going to use for my project, I am going to user the machine learning approach which will include these steps.

1. Data collection
2. Data cleaning
3. Model building
4. Evaluate
5. Model deployment

These steps will be used to complete this project where I will use the data set I have chosen to complete the data collection step of this model, I will then open the dataset in R studio to then check if the data has any null data contained in the dataset and use methods to replace the data so that there are no errors, in the next step I will then write the code for each of my chosen models and start running them to make predictions on which accounts are at risk of becoming default. I will then use code in R studios to make graphs to show where there could be errors in the data made from the predictions to make a good evaluation of the models used to determine which model is going to be my product. Then for the final step I will upload my code on to git hub through my process to make sure that there is come records of time and that my final product is outlined for other users to take and deploy onto their systems, in these last couple steps I will need to make a presentation about my product which will include my evaluation of each model and a dashboard to make it presentable.

# Data

## Missing values

In this section I will look at the dataset taken for my project and look though it for missing data to then fill the spots in via KNN. This is done to make the predictions more accurate since there won't be any null data in the dataset

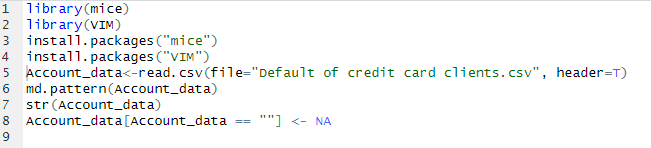


Figure code for missing values

This is the code I used to allow R studio to gain access to the data set and check for any null data. This was done by downloading two libraries these are mice and VIM, the library mice is used to use the md.pattern function to check the data set for any null values, the second library is VIM this is used for KNN predictions to fill these missing values. Line 5 was used to allow R to read the data set making it a variable.

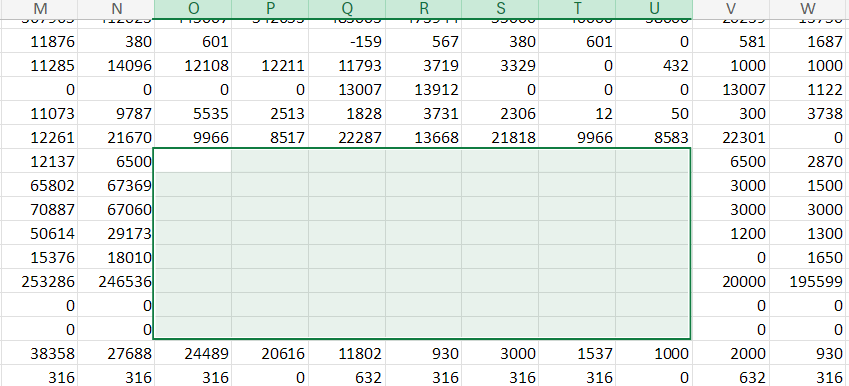


Figure picture of dataset showing missing data

Looking at the dataset there is some missing values but they are not in the correct format since we will need them to be N/A to make them show up while using the md.pattern function. To fix this issue in the data set we will need to change this data to a correct format, this was done by this line of code:



Figure code for making null values

After doing this we can see that the data has been changed.

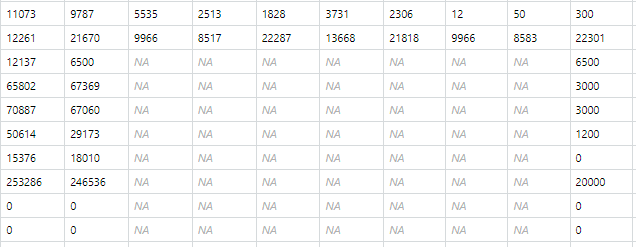


Figure dataset showing null values

After this we can use md.pattern to see where the data is missing in the data set.



Figure code to find missing values

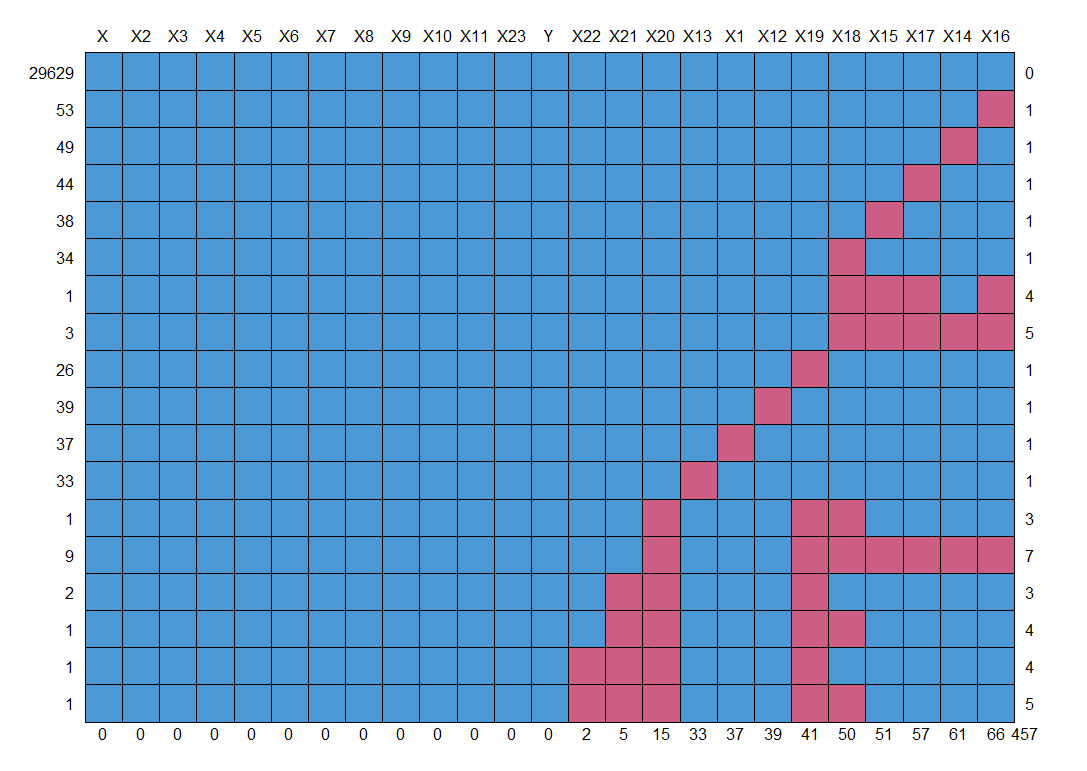


Figure output of md.pattern code

Now we can start filling in these null data.



Figure KNN method code

Line 9 is the code that is used to fill in the null data using KNN, I had to create a new variable to which is the new imputed data created. After this we can then look at the summary of the data to see if the data has been changed:

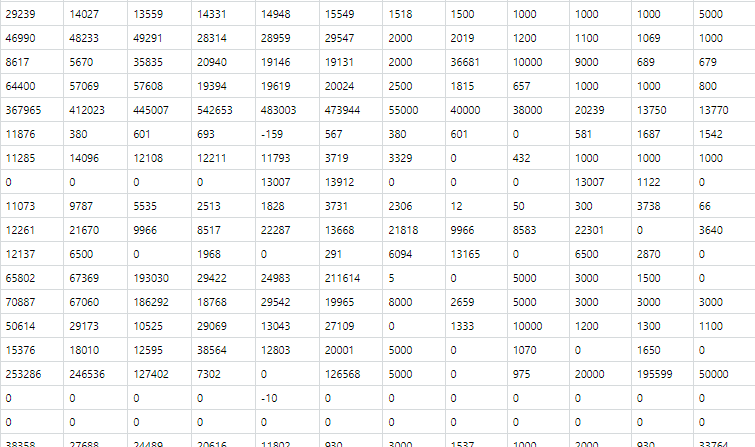


Figure photo of dataset showing null values removed and filled

But the best way that this can be easily seen is by using md.pattern to make a graph to see any null data:

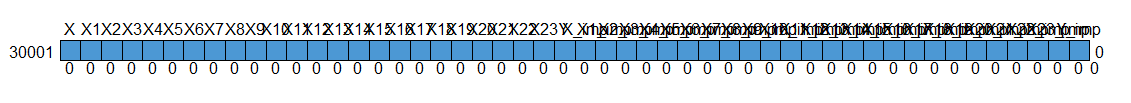


Figure md.pattern output after data imputation

now after this we will need to write this new imputed data into the data set for us to call back to on a later date. This was done by the code below which created a new file called Default of credit card clients cleaned.csv:



Figure code used to save changed dataset to an excel file

## Finding outliers

Outliers are data in the dataset that is data that is outside the interquartile ranges, these are normally removed to allow for improved accuracy and readability while doing data analysis. Doing this I implemented a for loop to make finding outlier more efficient since I will not have to run the same code over and over again for each column in my dataset. This was done with this code:

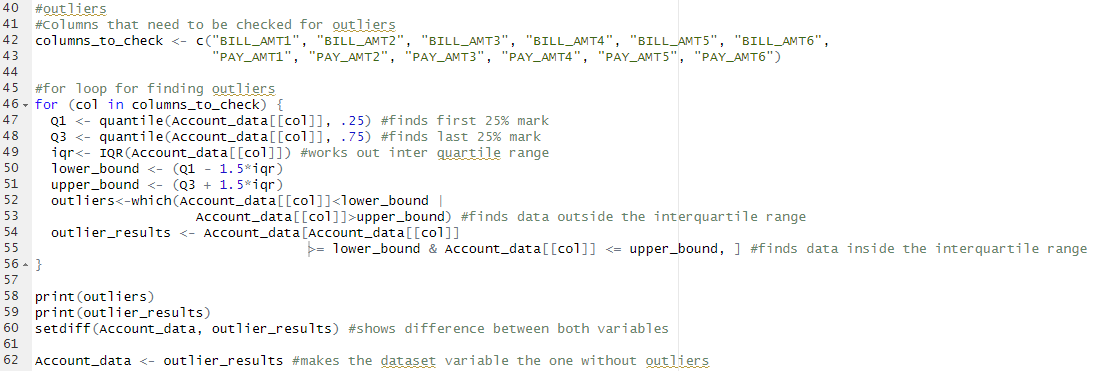


Figure code for finding outliers

In this code I have split it into different sections the first one is a variable that I have made which contains all the column names I want to investigate for outliers, this was done to allow for easier use of calling the columns in the for loop, the next section is the for loop which checks for each column in the dataset that I have called above to look for their interquartile range and the median to be able to work out which values are outside the expected range/values that are extreme, then using this in the for loop to be able to output these outliers for use to see the specific data as shown below:

A screenshot of a computer screen

Description automatically generated

Figure output showing outliers

this is all the data that is extreme and can be identified as an outlier, using this I can see that there are 1958, which is a large amount of data that can be considered an outlier so I decided to remove them from the data set this was done by using code at line 62 this makes my outliers are removed from my dataset by using a variable. I made in the for loop at line 54 which uses the interquartile ranges to find the data within the range, the code at line 62 uses the variable that contains the data within the range and makes that my new data variable, after this I then saved the dataset as cleaned 2.0.

# Results

## Logistic regression

A screenshot of a computer program

Description automatically generated

Figure implementation of logistic regression

Logistic regression is a supervised machine learning model which is mostly used for predicting classification problems. This is mostly used for fraud detection, disease prediction, churn prediction. This model only predicts true or false or 1 and 0. Which is useful to me since my y column in the dataset is binary.

As shown in figure 20 I used multiple different techniques to develop this model, the first technique that I used was data splitting this was done so that I have 2 different variables containing data one for testing and one for training, the training data is used while developing the logistic regression to train the model to find what data I want to predict, and the test one is used to test the model accuracy to see which out of the 5 models are the best, this was done by using the confusion matrix function to give statistics of the model as seen in figure 21. Anther technique that I utilize is data conversion, this was done since when it came to using confusion matrix, r studio would output an error message saying that the 2 variables used are not of the same factor levels, to fix this I make sure that the variables for test and train are the same factor level this was done by telling r that the default.payment.next.month is 1 and 0. These methods were utilized in all the models developed. I also used the predict function to predict the outcome for the test variable using the trained model.

A screenshot of a computer

Description automatically generated

Figure logistic regression accuracy

In figure 21 this is the output of using the confusion matrix on this model, this shows that the accuracy for predicting if an account is default is 81.1%, also if we look at the kappa score this model was given it tells us that it indicates fair agreement.

## KNN

A screenshot of a computer program

Description automatically generated

Figure implementation of KNN

The KNN model is a supervised machine learning algorithm that is used for classification, this model will use all data points to find data on a similarity measure. In the code in figure 22 I have used the same technique as talked about above for logistic regression being, data splitting, factorizing and even using predict and confusion matrix, this model differs by using a label when splitting the data set, the label is my y column what I am trying to predict when developing this model the label is used to make sure that the y column for the train and test variables that are used for the prediction and evaluation have the correct y column to work from.

A screenshot of a computer

Description automatically generated

Figure KNN accuracy

In figure 23 this is the output of the confusion matrix for the KNN model, as we can see that the model has an overall accuracy of 79.35% and a kappa score of 0.3108 which shows that the model has a fair agreement showing some predictive power.

## Naïve Bayes

A screenshot of a computer code

Description automatically generated

Figure implementation of naive Bayes

Naïve Bayes is a machine learning model that uses the bayers theorem to see independence between features in a dataset. In figure 24, it shows the model for naïve bayes which uses the same techniques as talk about for logistic regression, except for this model uses a function to normalize the data to make the features of the data have the same scale for model prediction. When it comes to the model development I used 2 way to develop using raw and class, class is used for predicting class, w0.3hich is when it makes a hard decision one or the other, where raw is used to see what the possibilities the different outcomes are.

A screenshot of a computer

Description automatically generated

Figure Naive Bayes accuracy

In figure 25 we can see the output for the confusion matrix for naïve Bayes model developed, this shows us the model has an accuracy of 69.41% and a kappa of 0.3001 which shows some fair agreement same as knn.

## Random Forest

A screenshot of a computer program

Description automatically generated

Figure implementation of random forest

Random forrest is a very similar model to decision tree since the both use trees, randome forest uses multiple different trees though its model so runnin it is quite slow which could be a bad thing for the industy since time is valuable and this models stats are similar to other models that take less time. When it comes the development of this model as seen in figure 26 I used the same techniques and function as the other models, and the traing of the model was the simplest since all you do is use the random forest function which is apart of the random forest package though this model is great for visuals since there is a simple way to plot the results on a graph using the plot function.

A screenshot of a computer

Description automatically generated

Figure random forest accuracy

When it comes to the statistics for the random forest model I used the confusion matrix like the other models and the output can be seen in figure 27, we can see that the model accuracy is at 81.72% and the kappa score being 0.3736 which like other models shows some predictive power and fair agreement.

## Decision tree

A screenshot of a computer code

Description automatically generated

Figure implementation of decision tree

When it comes to the decision tree model it uses one tree instead of many like random forest making it faster but with more complicated datasets the accuracy should be lower. As we can see in figure 28 the code used for developing the model shows that the same techniques are used as data splitting and factorizing, training this model is similar to random forest though instead of using randomForest function I used rpart for training the model and used other function like confusion matrix to work out the stats of this model.

A screenshot of a computer

Description automatically generated

Figure decision tree accuracy

When it comes to the stats of the model we can see them in figure 29 which tells us that the model has an accuracy of 82.07% which tells me that this model has the best accuracy and that the kappa score also has a similar score to other models showing that the model has some predictive power.

|  |  |  |
| --- | --- | --- |
| Model | Accuracy % | Kappa |
| Logistic Regression | 81.12% | 0.2993 |
| KNN | 79.35% | 0.3108 |
| Naïve Bayers | 69.42% | 0.3001 |
| Random Forest | 81.72% | 0.3736 |
| Decision tree | 82.07% | 0.3536 |

Figure table of all accuracy scores

# Conclusion

When it comes the my product I will chose the model with the best accuracy at predicting if an account is at risk of becoming default, as we can see in figure 30 which shows the accuracy of all the model developed we can see that the model with the best accuracy is decision tree which comes at an accuracy of 82.07% which is very similar to the other models other than naïve Bayes which shows us that this type of model is not well suited for this task, though the other models accuracy is similar decision tree has the highest for this task, it is also one of the easiest to develop and implement in the industry. Though other models could be better depending on what the task at hand is. If we look at other metrics shown by the confusion matrix we can see that the kappa score is very similar between all the models developed so all the models have a fair agreement, which tells us that there are other models that could be developed that could handle this task better but all these models have still some levels of predictability. If I was to extend this project I would look at making some hybrid machine learning models to see if they have a better accuracy for prediction and a stronger prediction power via kappa score.

# Appendix

## Risk register

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Risk Description** | **Likelihood** | **Impact** | **Owner** | **Mitigation** | **Status** |
| **1** | **Loss of data** | Low | High | **Jack** | **Save data to cloud** | **Open** |
| **2** | **Software accessibility** | Low | High | **Jack** | **Download software** | **Closed** |
| **3** | **Miss use of data** | Medium | High | **Jack** | **Learn about legistrations and how to follow them** | **Open** |
| **4** | Dataset could be full of errors | Medium | High | Dataset owner/ Jack | Inspect data set to see if there are any errors and send feedback to the owner to make appropriate changes | Open |
| **5** | Product Implementation | Medium | High | Police Force | The Banks would need to check if they could implement my product into their system | OPEN |

## Bibliography

Dahiwade, D., Patle, G. and Meshram, E. (2019) Designing disease prediction model using machine learning approach. 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, pp. 1211–1215. Available at: https://doi.org/10.1109/ICCMC.2019.8819782 (Accessed: 03/02/25)

Grampurohit, S. and Sagarnal, C. (2020) Disease prediction using machine learning algorithms. 2020 International Conference for Emerging Technology (INCET), Belgaum, India, pp. 1–7. Available at: https://doi.org/10.1109/INCET49848.2020.9154130 (Accessed: 03/02/25)

Islam, S.R., Eberle, W. and Ghafoor, S.K. (2018) Credit default mining using combined machine learning and heuristic approach. ArXiv, abs/1807.01176.

Labellerr. (2022) Machine learning steps: A complete guide for beginner in ML. [online] Available at: https://www.labellerr.com/blog/machine-learning-steps-a-complete-guide-to-the-ml-process/

Lontsi Saadio, C., Adoni, W.Y.H., Aworka, R., Zoueu, J.T., Kalala Mutombo, F. and Kimpolo, C.L.M. (2022) Crops Yield Prediction Based on Machine Learning Models: Case of West African Countries. SSRN Electronic Journal. doi: https://doi.org/10.2139/ssrn.4003105

López Steinmetz, L.C. et al. (2024) Machine learning models predict the emergence of depression in Argentinean college students during periods of COVID-19 quarantine. Frontiers in Psychiatry, 15. Available at: https://doi.org/10.3389/fpsyt.2024.1376784

Luna, J.C. (2020) Python vs R for Data Science: Which Should You Learn? [online] Datacamp.com. Available at: https://www.datacamp.com/blog/python-vs-r-for-data-science-whats-the-difference?dc\_referrer=https%3A%2F%2Fwww.google.com%2F (Accessed: 24 Feb. 2025)

Mosavi, A., Ozturk, P. and Chau, K.-w. (2018) Flood prediction using machine learning models: Literature review. Water, 10(11), p. 1536. Available at: https://doi.org/10.3390/w10111536 (Accessed: 03/02/25)

Mulder, P. (2018) Heuristic Method, a problem-solving method | ToolsHero. [online] toolshero. Available at: https://www.toolshero.com/problem-solving/heuristic-method/

Pamporaki, C. et al. (2023) Machine learning models for prediction of metastatic pheochromocytoma and paraganglioma: a cross-sectional cohort study. The Lancet. Digital Health, 5(9), pp. e551–e559. Available at: https://doi.org/10.1016/S2589-7500(23)00094-8

Perović, Marija et al. (no date) Machine Learning Models for Predicting the Ammonium Concentration in Alluvial Groundwaters. Environmental Modeling & Assessment, 26(2), pp. 187–203. Available at: https://doi.org/10.1007/s10666-020-09731-9

Wang, Y. et al. (2023) Prediction of TBM operation parameters using machine learning models based on BPSO. Advanced Engineering Informatics, 56. Available at: https://doi.org/10.1016/j.aei.2023.101955

Yeh, I. (2009) [Dataset] Default of Credit Card Clients. UCI Machine Learning Repository. Available at: https://doi.org/10.24432/C55S3H

## Use of Generative Artificial Intelligence (GAI) Statement

|  |  |  |  |
| --- | --- | --- | --- |
| **Appendix** | **Generative AI Tool (e.g. ChatGPT)** | **How generative AI Tool was used** | **Reference** |
| A | ChatGPT | Asked it to convert cites form articles into Leeds beckets havard style | OpenAI. (2023). ChatGPT (Sept 12 version) [Large language model]. https://chat.openai.com/chat |
| B | ChatGPT | why does knn use a label can you remove it and get the same result | OpenAI. (2023). ChatGPT (Sept 12 version) [Large language model]. https://chat.openai.com/chat |
| C | chatGPT | Asked it to convert citations into the correct style using an example | OpenAI. (2023). ChatGPT (Sept 12 version) [Large language model]. https://chat.openai.com/chat |

## Monthly reports

A screenshot of a black screen

AI-generated content may be incorrect.

Figure November monthly report

A screenshot of a black screen

AI-generated content may be incorrect.

Figure December monthly report

A screenshot of a black screen

AI-generated content may be incorrect.

Figure January monthly report

A black and white list of tasks

AI-generated content may be incorrect.

Figure February monthly report

A screenshot of a black screen

AI-generated content may be incorrect.

Figure March monthly report

These monthly reports shows that I have been keeping up with my project since there are reports for every month that shows what work has been done and areas that are going to be worked on, helps show time scale of project.

## Github

<https://github.com/JacSkel/Project-Module.git>

A screenshot of a computer

AI-generated content may be incorrect.

Figure photo of github page

Github is where all my coding and files are saved, this is helpful since it is all stored on the cloud and can be made public for people to access to, the commits also shows that the project has been worked on over time.

## Dashboard

A screenshot of a computer

AI-generated content may be incorrect.

Figure Photo of dashboard

Dashboard shows the stats of each model from the confusion matrix function, I used a splicer to be able to filter this data on the dashboard to each model for more insight for specific models, the main bar chart shows the different between each model visually, when using the splicer this chart gets filtered out to the specific model the user wants to look at, that pie chart shows the accuracy of each model in a simple way this shows how similar the models accuracy is, I added labels for each section to make it easier to understand the percentage of accuracy. I added a funnel chart to shows the detection rate of all the models to see what models had the most detections. I added some buttons on top of the dashboard to show the data is a big visual for easier use, using the splicer this data can be changed to specific models, default is the top of the table. I added some format conditions to the table so that any one who wants to look at the table for all the data there is an easier way to understand which data is the best for the column.

## Video

<https://youtu.be/UD0_TPo7-88>

youtube video going over project

## Code

#load libraries

library(mice)

library(VIM)

install.packages("mice")

install.packages("VIM")

#import dataset

Account\_data<-read.csv(file="Default of credit card clients.csv", header=T) #original dataset

Account\_data<-read.csv(file="Default of credit card clients cleaned.csv", header=T) #dataset after missing values and column names, version 1

Account\_data<-read.csv(file="Default of credit card clients cleaned 2.0.csv", header=T) #dataset after removal of outliers, version 2

#find missing value

md.pattern(Account\_data) #shows graph of missing values

str(Account\_data)

#clean data

Account\_data[Account\_data == ""] <- NA #replaces blank data to NA for data imputing

Account\_data\_imputed <- kNN(Account\_data, variable = names(Account\_data), k = 5) #fills NA spaces using KNN

summary(Account\_data\_imputed)

md.pattern(Account\_data\_imputed)

#finding number of missing values percentage

missing\_percentage <- (colSums(is.na(Account\_data)) / nrow(Account\_data)) \* 100 #works out percentage of each missing data per variable

print(missing\_percentage)

#removing false column names

correct\_col<-names <- as.character(Account\_data[1, ]) #stores actual column names e.g. ID

Account\_data <- Account\_data[-1, ] #removes the actaul column names from dataset

colnames(Account\_data) <- correct\_col #makes the column names the actual ones

head(Account\_data)

str(Account\_data)

#removing columns made by KNN data imputing

Account\_data <- Account\_data[, -c(26:50)] #removes all flase spaces added after KNN data impution

#rite cleaned data set to file

write.csv(Account\_data, "Default of credit card clients cleaned.csv", row.names = FALSE) #write version 1 of dataset

write.csv(Account\_data, "Default of credit card clients cleaned 2.0.csv", row.names = FALSE) #write version 2 of dataset

#outliers

#Columns that need to be checked for outliers

columns\_to\_check <- c("BILL\_AMT1", "BILL\_AMT2", "BILL\_AMT3", "BILL\_AMT4", "BILL\_AMT5", "BILL\_AMT6",

"PAY\_AMT1", "PAY\_AMT2", "PAY\_AMT3", "PAY\_AMT4", "PAY\_AMT5", "PAY\_AMT6")

#for loop for finding outliers

for (col in columns\_to\_check) {

Q1 <- quantile(Account\_data[[col]], .25) #finds first 25% mark

Q3 <- quantile(Account\_data[[col]], .75) #finds last 25% mark

iqr<- IQR(Account\_data[[col]]) #works out inter quartile range

lower\_bound <- (Q1 - 1.5\*iqr)

upper\_bound <- (Q3 + 1.5\*iqr)

outliers<-which(Account\_data[[col]]<lower\_bound | Account\_data[[col]]>upper\_bound) #finds data outside the interquartile range

outlier\_results <- Account\_data[Account\_data[[col]] >= lower\_bound & Account\_data[[col]] <= upper\_bound, ] #finds data inside the interquartile range

}

print(outliers)

print(outlier\_results)

setdiff(Account\_data, outlier\_results) #shows difference between both variables

Account\_data <- outlier\_results #makes the dataset variable the one without outliers

#load libraries

library(mlbench)

library(caret)

install.packages("mlbench")

install.packages("caret")

#insert data set and EDA

Account\_data<-read.csv(file="Default of credit card clients cleaned 2.0.csv", header=T)

#splitting data for training

indexes<-sample(1:nrow(Account\_data), 4/5\*nrow(Account\_data))

train<-Account\_data[indexes,] #80%

test<-Account\_data[-indexes,] #last 20%

#code used to make sure variables are the same factor levels being binary 1 & 0

train$default.payment.next.month <- factor(train$default.payment.next.month, levels = c(

"0", "1"))

test$default.payment.next.month <- factor(test$default.payment.next.month, levels = c(

"0", "1"))

#model development

caret\_glm\_mod = train(

form = default.payment.next.month ~ .,

data = train,

trControl = trainControl(method = "cv", number = 5),

method = "glm",

family = "binomial"

)

#model evaluation

#creates prediction using model and test data

predicted\_test <- predict(caret\_glm\_mod, newdata = test)

#compare results from prediction and test

confusionMatrix(predicted\_test, test$default.payment.next.month, positive = "1")

#install libraries

library(class)

library(caret)

install.packages("class")

install.packages("caret")

#loading data set

Account\_data <- read.csv(file="Default of credit card clients cleaned 2.0.csv", header=T)

#normalizing data

normalize <- function(x) {

return ((x -min(x)) / (max(x) - min(x)))

}

acc\_norm <- as.data.frame(lapply(Account\_data[2:25], normalize))

#splitting data for training

indxTrain <- createDataPartition(y = Account\_data$default.payment.next.month,p = 0.8,

list = FALSE)

acc\_train <- acc\_norm[indxTrain,]

acc\_train\_label<-Account\_data$default.payment.next.month[indxTrain]

acc\_test <- acc\_norm[-indxTrain,]

acc\_test\_label<-Account\_data$default.payment.next.month[-indxTrain]

#code used to make sure variables are the same factor levels being binary 1 & 0

acc\_train\_label <- factor(acc\_train\_label, levels = c("0", "1"))

acc\_test\_label <- factor(acc\_test\_label, levels = c("0", "1"))

#model development using caret

acc\_train\_2 <- acc\_train

acc\_train\_2$default.payment.next.month <- acc\_train\_label

acc\_test\_2 <- acc\_test

acc\_test\_2$default.payment.next.month <- acc\_test\_label

ctrl <- trainControl(method = "cv", number = 5)

knnFit <- train(default.payment.next.month ~ ., data = acc\_train\_2, method = "knn",

trControl = ctrl)

knnFit

#model evaluation

knnPredict <- predict(knnFit,newdata = acc\_test\_2)

knnPredict

confusionMatrix(knnPredict, acc\_test\_label)

#install libraries

library("e1071")

library(caret)

install.packages("caret")

install.packages("e1071")

#loading data set

Account\_data <- read.csv(file="Default of credit card clients cleaned 2.0.csv", header=T)

#normalizing data

convert\_counts <- function(x) {

x <- ifelse(x > 0, 1, 0)

x <- factor(x, levels = c(0, 1), labels = c("No", "Yes"))

2

return(x)

}

#splitting data for training

ind<-createDataPartition(Account\_data$default.payment.next.month, p=0.7, list=F)

train<-Account\_data[ind,]

test<-Account\_data[-ind,]

#code used to make sure variables are the same factor levels being binary 1 & 0

train$default.payment.next.month <- factor(train$default.payment.next.month, levels = c(

"0", "1"))

test$default.payment.next.month <- factor(test$default.payment.next.month, levels = c(

"0", "1"))

#model development

acc\_classifier <- naiveBayes(train[, -ncol(train)], train$default.payment.next.month)

acc\_classifier

predicted <- predict(acc\_classifier, train[, ncol(train)], type="class")

predicted[1:100]

predicted<-predict(acc\_classifier, train[, -ncol(train)], type="raw")

predicted[1:100]

#model evaluation

acc\_test\_pred <- predict(acc\_classifier, test)

cm<-confusionMatrix(acc\_test\_pred, test$default.payment.next.month)

cm

#load libraries

library(randomForest)

library(caret)

install.packages("randomForest")

install.packages("caret")

#add data set

Account\_data<-read.csv(file="Default of credit card clients cleaned 2.0.csv", header=T)

str(Account\_data)

summary(Account\_data)

#split data for training

indexes<-sample(1:nrow(Account\_data), 4/5\*nrow(Account\_data))

train<-Account\_data[indexes,] #80%

test<-Account\_data[-indexes,] #last 20%

#code used to make sure variables are the same factor levels being binary 1 & 0

train$default.payment.next.month <- factor(train$default.payment.next.month, levels = c(

"0", "1"))

test$default.payment.next.month <- factor(test$default.payment.next.month, levels = c(

"0", "1"))

#Model Development

Default.rf=randomForest(default.payment.next.month ~ . , data = train)

Default.rf

#plot random forest

plot(Default.rf)

#model evaluation

pred<-predict(Default.rf, test)

confusionMatrix(data = pred, reference = test$default.payment.next.month)

#load libraries

library(ggplot2)

library(rpart)

library(rpart.plot)

library(caret)

install.packages("ggplot2")

install.packages("rpart")

install.packages("rpart.plot")

install.packages("caret")

#load data set

Account\_data<-read.csv(file="Default of credit card clients cleaned.csv", header=T)

str(Account\_data)

summary(Account\_data)

#splitting data for training

indexes<-sample(1:nrow(Account\_data), 4/5\*nrow(Account\_data))

train<-Account\_data[indexes,] #80%

test<-Account\_data[-indexes,] #last 20%

#code used to make sure variables are the same factor levels being binary 1 & 0

train$default.payment.next.month <- factor(train$default.payment.next.month, levels = c(

"0", "1"))

test$default.payment.next.month <- factor(test$default.payment.next.month, levels = c(

"0", "1"))

#model development

cart\_fit<-rpart(default.payment.next.month ~ ., data = train, method = "class")

summary(cart\_fit)

#plot a decision tree

rpart.plot(cart\_fit)

#model evaluation

Account\_pred <- predict(cart\_fit,test,type = "class")

confusionMatrix(data = Account\_pred, reference = test$default.payment.next.month)